UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/537,571	06/03/2005	Hoi Yeung Chan	YOR920020206US1 (8728-587	4559

46069 7590 12/31/2008

F. CHAU & ASSOCIATES, LLC
130 WOODBURY ROAD
WOODBURY, NY 11797

EXAMINER	
BROWN JR, NATHAN H	
ART UNIT	PAPER NUMBER
2129	
MAIL DATE	DELIVERY MODE
12/31/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/537,571
Filing Date: June 03, 2005
Appelant(s): CHAN ET AL.

Nathaniel T. Wallace
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed October 20, 2008

appealing from the Office action mailed May 23, 2008.

(1) Real Party in Interest

The real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

IBM, "WebSphere Application Server Enterprise Services Business Rule Beans (BRBeans), "2001

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1, 3-10, 13, 16-18, 20, and 36 are rejected under 35 U.S.C. 101 because the claimed invention is not directed to a practical application and violates the doctrine of preemption. Amended independent claim 1 recites a “computer readable medium embodying a program of instructions executable by a processor to perform a method deriving knowledge from parameters and data” producing a final result of “outputting the knowledge derived by the inference to the program of instructions”. The recited “knowledge” derived “from parameters and data” is considered to be a mathematical abstraction of logical reasoning. The method for “deriving knowledge from parameters and data” is considered

to be an algorithm. The final result of "outputting the knowledge derived by the inference to the program of instructions" does not provide a specific and substantial result such as the final share price, momentarily fixed for recording and reporting, in *State Street;* as the knowledge recited does not represent specific and substantial entities and relationships in a real-world problem domain.

Further, since knowledge derived from "parameters" and "data" can represent any conceivable thing or situation, claim 1 violates the doctrine of preemption by seeking patent protection for the application of the claimed method to every substantial application. Claims 3-10, 13, 16-18, and 20 merely provide further algorithmic limitation of claim 1, and thus do not cure the deficiency of claim 1. Therefore claims 1, 3-10, 13, 16-18, and 20 are considered non-statutory under 35 U.S.C. 101.

The claims fail to provide a tangible result, and there must be a practical application, by either

1) transforming (physical thing) or
2) by having the FINAL RESULT (not the steps) achieve or produce
   a useful (specific, substantial, AND credible),
   concrete (substantially repeatable/non-unpredictable), AND
   tangible (real world/non-abstract) result.

A claim that is so broad that it reads on both statutory and non-statutory subject matter, must be amended. A claim that recites a computer that solely calculates a mathematical formula is not statutory.

However, the portions of the opinions in State Street and AT&T relying solely on a "useful, concrete and tangible" result analysis *should no longer be relied on*. Ex parte Bilski, Appeal No. 2007-1130 (Fed. Cir. October 30, 2008.

The court has said that there's a two-pronged test to determine whether a software of business method process patent is valid: (1) it is tied to a particular machine or apparatus, or (2) it transforms a particular article into a different state or thing. In other words, pure software or business method patents that are neither tied to a specific machine nor change something into a different state are not patentable. Ex parte Bilski, Appeal No. 2007-1130 (Fed. Cir. October 30, 2008). The courts have also held that a claim may not preempt ideas, laws of nature or natural phenomena. The concern over preemption was expressed as early as 1852. See Le Roy v. Tatham, 55 U.S. (14 How.) 156, 175 (1852) ("A principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right."); Funk Bros. Seed Co. v. Kalo Inoculant Co., 333 U.S. 127, 132, 76 USPQ 280, 282 (1948).

Accordingly, one may not patent every "substantial practical application" of an idea, law of nature or natural phenomena because such a patent "in practical effect would be a patent on the [idea, law of nature or natural phenomena] itself." "Here the "process" claim is so abstract and sweeping as to cover both known and unknown uses of the BCD to pure-

binary conversion. The end use may (1) vary from the operation of a train to verification of drivers' licenses to researching the law books for precedents and (2) be performed through any existing machinery or future-devised machinery or without any apparatus." Gottschalk v. Benson, 409 U.S. 63, 71-72, 175 USPQ 673, 676 (1972).

> [In Abele], we held unpatentable a broad independent claim reciting a process of graphically displaying variances of data from average values. *Abele*, 684 F.2d at 909. **That claim did not specify any particular type or nature of data; nor did it specify how or from where the data was obtained or what the data represented**. *Id.*; ... In contrast, we held one of Abele's dependent claims to be drawn to patent-eligible subject matter where it specified that "said data is X-ray attenuation data produced in a two dimensional field by a computed tomography scanner." *Abele*, 684 F.2d at 908-09. This data clearly represented physical and tangible objects, namely the structure of bones, organs, and other body tissues. Thus, the transformation of that raw data into a particular visual depiction of a physical object on a display was sufficient to render that more narrowly-claimed process patent-eligible.
>
> … So long as the claimed process is limited to a practical application of a fundamental principle to transform **specific** data, and the claim is limited to a **visual depiction that represents specific physical objects or substances**, there is no danger that the scope of the claim would wholly pre-empt all uses of the principle.
>
> This court and our predecessor court have frequently stated that adding a data-gathering step to an algorithm is insufficient to convert that algorithm into a patent-eligible process. *E.g.*, *Grams*, 888 F.2d at 840 (step of "deriv[ing] data for the algorithm will not render the claim statutory"); *Meyer*, 688 F.2d at 794 ("[data-gathering] step[s] cannot make an otherwise nonstatutory claim statutory"). … **A requirement simply that data inputs be gathered—without specifying how—is a meaningless limit** on a claim to an algorithm because every algorithm inherently requires the gathering of data inputs. *Grams*, 888 F.2d at 839-40. Further, the inherent step of gathering data can also fairly be characterized as **insignificant extra-solution activity**. *See Flook*, 437 U.S. at 590. (See In re Bilski, 88 USPQ2d 1397-1398, emphasis added)

The Courts have found that subject matter that is not a practical application or use of an idea, a law of nature or a natural phenomenon is not patentable. As the Supreme Court has

made clear, " [a]n idea of itself is not patentable," *Rubber-Tip Pencil Co. v. Howard,* 20 U.S. (1 Wall.) 498, 507 (1874); taking several abstract ideas and manipulating them together adds nothing to the basic equation. In re Warmerdam, 31 USPQ2d 1754 (Fed. Cir. 1994).

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

Claims 1-33 are rejected under 35 U.S.C. 102(a) as being anticipated by ***IBM,*** "WebSphere Application Server Enterprise Services Business Rule Beans (BRBeans)", 2001.

Regarding claim 1. (Currently Amended) A computer readable medium (not further defined) embodying a program of instructions executable by a processor to perform a method for deriving knowledge from parameters and data (*see* p. 8, §Getting Started with BRBeans, *Examiner interprets steps 1 and 2 to install a program of instructions on a computer readable medium e.g., a disk drive, of an AE server to be embodying a program of instructions executable by a processor.*), the method comprising:

passing the parameters (not further defined) to an externalized inferencing component (not further defined) upon executing a trigger point in the program of instructions (*see* pp. 26-27, §Situational Trigger Point - an example, code fragment, *Examiner interprets: a classifier rule to be an externalized inferencing component (Business Rule Bean (see pp. 2-4)) the data of the externalized inferencing component to be the "Customer cust" object, "TriggerPoint tp" to be a trigger point, and "Object [] classifierPlist" to comprise parameters passed by a trigger point within the program of instructions to perform an inference, and "result = tp.triggerSituational(this, classifiedPlist, classifierPlist, classifiedRule, classifierRuleName)" to be the point where execution occurs.*);

evaluating, by the externalized inferencing component, the data comprising a set of rules to be interpreted of an externalized inferencing component in the presence of against the parameters to perform an inference external to the program of instructions passed by a trigger point within the program of instructions to perform an inference (*see* §The BRBeans framework – overview, BRBeans EJBs, p. 18 and §Runtime behavior, p. 32-33, *Examiner interprets the "WebSphere application server in which the BRBeans EJBs are installed" to comprise relational databases in which all rule classes, entity classes, and trigger point*

classes are implemented in relational database tables. Thus any evaluation of data comprising a set of externalized inferencing rules against parameters to perform an inference are simply queries applied to the associated tables defined in WebSphere.), wherein the externalized inferencing component is in communication with the program of instructions (*see* p. 20, *Examiner interprets the "Firing Location" parameters to govern how the externalized inferencing component is in communication with the program of instructions.*), wherein the inference is a derivation of the knowledge (*see* p. 45, §Writing your own strategies, steps 1-3, *Examiner interprets the "firing strategy fires the rules found by the finding strategy, possibly modified by the filtering strategy…gathers up the results of the individual rules and these results are passed to the combining strategy" to implement derivation of the knowledge through firing a rule (i.e., inferencing).*);

storing the knowledge derived by the inference with the data (*see* §The BRBeans framework – overview, BRBeans EJBs, p. 18 and §Runtime behavior, p. 32-33, *Examiner interprets the "WebSphere application server in which the BRBeans EJBs are installed" to comprise relational databases in which all rule classes, entity classes, and trigger point classes are implemented in relational database tables. Thus knowledge*

derived (e.g., business situation knowledge) is necessarily stored in the database along with the data giving rise to the situational inference.); and

outputting the knowledge derived by the inference to trigger point of the program of instructions (*see* p. 45, §Writing your own strategies, step 3, *Examiner interprets passing the results of steps 1-3 "to the combining strategy" to be outputting the knowledge derived by the inference to the program of instructions "to produce the final result of the trigger".*).

Regarding claim 36. (Currently Amended) A system for executing a program of instructions in communication with an externalized inference component (*see* p. 2, *Examiner interprets "Websphere" to be system for executing a program of instructions in communication with an externalized inference component.*) comprising:

a memory device storing data, the program of instructions and the externalized inference component (*see* pp. 6-8, Examiner interprets a "relational databases…supported by BRBeans" to be memory devices storing data.);

a processor for receiving the data and executing the plurality of instructions and the externalized inference

component to perform a method for deriving knowledge from the data (*see* p. 47, "If these are not specified, then the name server used by the container in which the application is running is used. If the application is not running in a container, then localhost is used for the host name, and 900 is used for the port number.", *Examiner interprets the "name server" or the "localhost" to be computers comprising a processor for receiving the data and executing the plurality of instructions and the externalized inference component to perform a method for deriving knowledge from the data.*) comprising:

passing the parameters to an externalized inferencing component upon executing a trigger point in the program of instructions (*see* pp. 26-27, §Situational Trigger Point - an example, code fragment, *Examiner interprets: a classifier rule to be an externalized inferencing component (Buisness Rule Bean (see pp. 2-4)) the data of the externalized inferencing component to be the "Customer cust" object, "TriggerPoint tp" to be a trigger point, and "Object [] classifierPlist" to comprise parameters passed by a trigger point within the program of instructions to perform an inference, and "result = tp.triggerSituational(this, classifiedPlist,*

classifierPlist, classifiedRule,classifierRuleName)" to be the point where execution occurs.);

evaluating, by the externalized inferencing component, the data comprising a set of rules to be interpreted of an externalized inferencing component in the presence of against the parameters to perform an inference external to the program of instructions passed by a trigger point within the program of instructions to perform an inference (see), wherein the externalized inferencing component is in communication with the program of instructions (*see* p. 20, *Examiner interprets the "Firing Location" parameters to govern how the externalized inferencing component is in communication with the program of instructions*.), wherein the inference is a derivation of the knowledge (*see* p. 45, §Writing your own strategies, steps 1-3, *Examiner interprets the "firing strategy fires the rules found by the finding strategy, possibly modified by the filtering strategy…gathers up the results of the individual rules and these results are passed to the combining strategy" to implement derivation of the knowledge through firing a rule (i.e., inferencing)*.);

storing the knowledge derived by the inference with the data (*see* §The BRBeans framework - overview, BRBeans

EJBs, p. 18 and §Runtime behavior, p. 32-33, *Examiner interprets the "WebSphere application server in which the BRBeans EJBs are installed" to comprise relational databases in which all rule classes, entity classes, and trigger point classes are implemented in relational database tables. Thus knowledge derived (e.g., business situation knowledge) is necessarily stored in the database along with the data giving rise to the situational inference*.); and

outputting the knowledge derived by the inference to trigger point of the program

of instructions (*see* p. 45, §Writing your own strategies, step 3, *Examiner interprets passing the results of steps 1-3 "to the combining strategy" to be outputting the knowledge derived by the inference to the program of instructions "to produce the final result of the trigger"*.).

Regarding claim 3. (Currently Amended) *IBM* teaches the method of claim 1, wherein the data is stored in persistent memory external to the program of instructions (*see* pp. 6-8, §Database considerations, *Examiner interpret DB2, Oracle, Sybase, and*

Informix to store data in persistent memory external to the program of instructions.).

Regarding claim 4. (Currently Amended) *IBM* teaches the method of claim 1, wherein the externalized inferencing component includes at least one of a short term fact, an inference rule, an inference engine, a static variable mapping, a sensor, an effector, a long term fact, and a conclusion (*see* pp. 22-29, §Trigger Point Framework – overview).

Regarding claim 5. (Currently Amended) *IBM* teaches the method of claim 1, wherein the externalized inferencing component includes at least one of a short term fact component, an inference rule set component, an inference engine component, a static mapping component, a sensor component, an effector component, a long term fact component, and a conclusion component (*see* p. 22, §Trigger Point Framework – overview, *Examiner interprets "the BRBeans trigger point" to be part of the BRBeans externalized inferencing component set*.).

Regarding claim 6. (Previously Presented) *IBM* teaches the method of claim 2, wherein the externalized inferencing component is one of a consumer of data provided by an inferencing component,

a supplier of data provided by an inferencing component, and a combination thereof (*see* pp. 26-27, §Situational Trigger Point - an example, code fragment, *Examiner interprets: the externalized inferencing component to be the "Customer cust" object to be one of a consumer of data provided by an inferencing component*.).

Regarding claim 7. (Previously Presented) *IBM* teaches the method of claim 1, further comprising the step of associating the trigger point with the program of instructions (*see* pp. 26-27, §Situational Trigger Point - an example, code fragment, *Examiner interprets the method name called to associate the trigger point with the program of instructions, e.g., setCombiningStrategy in tp.setCombiningStrategy*.).

Regarding claim 8. (Original) *IBM* teaches the method of claim 4, wherein trigger points operate either synchronously or asynchronously (see p. 47, §As Of date, *Examiner interprets "in effect" to allow trigger points to fire rules synchronously and the setAsOfDate method on the TriggerPoint object to to allow trigger points to fire rules asynchronously*.).

Regarding claim 9. (Previously Presented) *IBM* teaches the method of claim 1, wherein the externalized inferencing component is a master inferencing component that employs at least one other externalized inferencing component (*see* pp. 26-27, §Situational Trigger Point - an example, code fragment, *Examiner interprets: "TriggerPoint tp" to be a master inferencing component that employs at least one other externalized inferencing component, a classifier rule (Buisness Rule Bean (see pp. 2-4))*.).

Regarding claim 10. (Previously Presented) *IBM* teaches the method of claim 1, wherein the externalized inferencing component employs an inferencing engine (*see* p. 45-46, §Writing your own strategies, *Examiner considers the TriggerPoint strategies (FindingStrategy, FilteringStrategy, FiringStrategy, and CombiningStrategy) to comprise an inferencing engine as they control rule firing*.).

Regarding claim 13. (Previously Presented) *IBM* teaches the method of claim 1, wherein the externalized inferencing component is composed of at least one inferencing subcomponent (*see* pp. 45-46, §Writing your own strategies, *Examiner interprets: "TriggerPoint tp" to be an inferencing component composed inferencing subcomponents, TriggerPoint strategies*

classes (FindingStrategy, FilteringStrategy, FiringStrategy, and CombiningStrategy).).

Regarding claim 16. (Previously Presented) *IBM* teaches the method of claim 1, further comprising sharing the externalized inferencing component by reference with at least one other externalized inferencing component (*see* p. 47, §The BRBeans Properties file, "When an application attempts to reference BRBeans EJBs, the code will first look for the brbPropertiesFile Java property. If this property is specified, then the names listed in that file are used to find the EJBs, overriding any EJB references that were specified in the container (if the application is running in a container). If the property is not specified, then BRBeans attempts to use the EJB references specified in the container.").

Regarding claim 17. (Previously Presented) *IBM* teaches the method of claim 1, wherein the externalized inferencing component performs method steps to one of create, update and delete another externalized inferencing component (*see* p. 34, §Rule Management APIs, IRule, "This is the interface used to access the object representing a business rule in BRBeans. It provides methods to read and update attributes of the rule, to

delete the rule, and to make a copy of the rule. The methods to create rules are on the IRuleFolder interface since you must always create a rule into a particular folder.", *Examiner interprets the "com.ibm.websphere.brb.mgmt package" to comprise externalized inferencing components*.).

Regarding claim 18. (Previously Presented) *IBM* teaches the method of claim 1, wherein an algorithm of the externalized inferencing component for performing the evaluation is shared by a plurality of externalized inferencing components (*see* pp. 26-27, §Situational Trigger Point - an example, code fragment, *Examiner interprets class "TriggerPoint" to be an* externalized inferencing component for performing the evaluation which can be shared by a plurality of externalized inferencing components by simply creating an instance, *"TriggerPoint tp", and then calling the instance, as in "result = tp.triggerSituational(this,…)", to perform an evaluation*.).

Regarding claim 20. (Previously Presented) *IBM* teaches the method of claim 1, further comprising providing an inference component management facility to administer externalized inferencing components, the administration including operations to create, retrieve, update, and delete (*see* p. 18, BRBeans Rule

Management Application, "The BRBeans Rule Management Application is implemented as a Java Application that runs stand-alone, remotely or locally to the BRBeans rule server. It is used to create, update, expire, and delete BRBeans Rules, and can also be used to interactively import and export BRBeans Rules from/to XML.", *Examiner interprets the "BRBeans Rule Management Application" to provide an inference component management facility to administer externalized inferencing components, the administration including operations to create, retrieve, update, and delete.*).

(10) Response to Argument

Applicant's arguments filed October 20, 2008 have been fully considered but they are not persuasive.

A. Rejection of Claims 1, 3-10, 13, 16-18, 20 and 36 Under 35 U.S.C. §101

Appellants' argue:

i. Claims 1, 3-10, 13, 16-18, 20 and 36

> Claims 1 and 36 claim, inter alia, "passing the parameters to an externalized inferencing component upon executing a trigger point in the program of instructions; evaluating, by the externalized inferencing component, the data comprising a set of rules to be interpreted against the parameters to

perform an inference external to the program of instructions, wherein the externalized inferencing component is in communication with the program of instructions, wherein the inference is a derivation of the knowledge; storing the knowledge derived by the inference with the data; and outputting the knowledge derived by the inference to trigger point of the program of instructions."

Such an application of an externalized inferencing component evaluating parameters passed by a trigger point to perform an inference is believed to be a practical application of a method (embodied in a computer readable medium (see Claim 1) and a system for executing a program of instructions in communication with an externalized inference component (see Claim 36)). Consider that in Diamond v. Diehr, 450 U.S. 175,209 USPQ 1 (1981), the Court noted, "when [a claimed invention] is performing a function which the patent laws were designed to protect (e.g., transforming or reducing an article to a different state or thing), then the claim satisfies the requirements of Section 101." Diehr, 450 U.S. at 192.

Examiner's response:

Examiner disagrees for at least the following reasons.

In the appeal brief, appellants fail to point to the disclosure to support the argument that there is a practical application and that the claims are not preemptive.

One may argue that the appellant solving all problems known to man. On pg. 2, the appellant discloses inferencing aspects as including facts, rules or conclusions, on pg. 7, the appellant states "rules" are not

those usually associated with AI but are ones used to make every day business decisions, on pg. 8, the appellant discusses frequent flyer miles and classifications thereof, on pg. 9-10, the appellant discuses people in China purchasing a camera. Therefore the examiner questions, what the practical application is.

Appellants' argue:

In Claims 1 and 36, the claimed computer processes perform inferencing external to a program of instructions, passing parameters by a trigger point for evaluation by an externalized inferencing component to perform the inferencing (external to the program of instructions). Passing parameters to an external inferencing component to perform inferencing which derives new knowledge from rules and knowledge (data and parameters), essentially as claimed in Claims 1 and 36, is believed to be a useful, tangible and concrete result of the application of the claimed limitations. For example, consider the following:

"[T]ransformation of data, representing discrete dollar amounts, by a machine through a series of mathematical calculations into a final share price, constitutes a practical application of a mathematical algorithm, formula, or calculation, because it produces 'a useful, concrete and tangible result' - a final share price momentarily fixed for recording and reporting purposes and even accepted and relied upon by regulatory authorities and in subsequent trades." State Street, 149 F.3d at 1373, 47 USPQ2d at 1601.

In formulating the rejection under 35 U.S.C. § 101, The Examiner states that "the inferencing components and their organization constitute a set of mathematical abstractions of data structure and control flow" while

> concluding that the claimed invention does not perform "a tangible transformation of real-world entities." (See Final Action, p. 20). Applicants respectfully disagree. As made clear in State Street, "transformation of data...through a series of mathematical calculations.., constitutes a practical application of a mathematical algorithm, formula, or calculation, because it produces 'a useful, concrete and tangible result'..." Id. Thus, according to State Street, the fact that the performed inferencing implements mathematical algorithms in order to transform rules and knowledge into new knowledge does not render the claimed invention unpatentable under 35 U.S.C. § 101. The claimed invention passes rules and knowledge (data and parameters) from an application to an external inferencing component, which utilizes mathematical algorithms to perform inferencing that transforms the rules and knowledge into new knowledge, which is then used in the application. Under State Street, this transformation is clearly a useful, tangible and concrete result of the application of the claimed limitations.
>
> Therefore, Claims 1 and 36 are believed to be directed towards statutory subject matter.

Examiner's response:

(A) Examiner asserts that "Passing parameters to an external inferencing component to perform inferencing which derives new knowledge from rules and knowledge (data and parameters), as essentially claimed, in Claims 1 and 36" is *not* a result, but merely a step in the program of instructions embodied in the claimed computer readable medium. As the appellants further clarify above, the:

> claimed invention passes rules and knowledge (data and parameters) from an application to an external inferencing component, which utilizes mathematical

> algorithms to perform inferencing that transforms the rules and knowledge into new knowledge, which is then used in the application...

Clearly, the "new knowledge" is not the end of the process and therefore not a tangible result of the program of instructions embodied in the claimed computer readable medium.

(B) Examiner contends that while appellants' have recited State Street they have missed the crux of State Street:

> "[T]ransformation of data, representing discrete dollar amounts, by a machine through a series of mathematical calculations into a final share price, constitutes a practical application of a mathematical algorithm, formula, or calculation, because it produces 'a useful, concrete and tangible result' - a final share price momentarily fixed for recording and reporting purposes and even accepted and relied upon by regulatory authorities and in subsequent trades." State Street, 149 F.3d at 1373, 47 USPQ2d at 1601.

It is not the mere "transformation of data...through a series of mathematical calculations" that State Street proffers as providing a tangible result, but the transformation of "of data, **representing discrete dollar amounts**,...through a series of mathematical calculations" ending up with "a **final share price** momentarily fixed for recording and reporting" [emphasis Examiner's]. The crux of State Street requires data *representing something* (in

the real-world) to be transformed into something else which *represents something* (in the real-world) and is final.

Appellants recite, merely, the transformation of data, per se, embodied as non-specific "rules and knowledge". Rules and knowledge, not directed to a discernable problem domain, must be considered mathematical abstractions (knowledge representation formalisms) of the type found in representational knowledge based artificial intelligence (AI definition, *see* e.g., Tychonievich et al., "THE DEFINITIONAL METHOD AND SIMPLICITY IN KNOWLEDGE BASED TOOLS", 1989) because they can play no part and have no effect in any practical problem solving involving specific things and their relationships. Tychonievich et al. provides an example of a practical application.

Appellants' argue:

> Turning now to the suggestion that the claims violated the doctrine of preemption: that is whether the claim would, in reality, preempt the use of a law of nature or abstract idea. While one may not patent a process that comprises every substantial practical application of an abstract idea, because such a patent in practical effect would be a patent on the abstract idea itself, the claims are clearly presented in terms of a computer readable medium (see Claim 1) and a system for executing a program of instructions in communication with an externalized inference component (see Claim 36). Accordingly, the claims are not

directed to mere abstract ideas but include limitations that are both concrete and tangible.

Examiner responds:

After reading claims 1, and 36 one is left asking, what problem doesn't the claimed invention solve? It would appear the applicant wants a patent to solve any problem known to man. Both the computer readable medium of claim 1 and the system of claim 36 are considered to be computer related manufactures reciting no data structure but reciting a functional program. Both the program of instructions of claim 1 and the system of claim 36 are considered to be combined with a non-paper tangible medium. Both the program of instructions of claim 1 and the system of claim 36 clearly do not transform a particular (physical) article from one state to another. At this juncture it must be shown that the claims produce a useful, concrete, and tangible result. If we are to focus on results, then we must (at some point) resort to the proviso of MPEP 2106 subsection IV.C.2:

> While abstract ideas, natural phenomena, and laws of nature are not eligible for patenting, methods and products employing abstract ideas, natural phenomena, and laws of nature to perform a real-world function may well be. In evaluating whether a claim meets the requirements of section 101, the claim must be considered as a whole to determine whether it is for a

> particular application of an abstract idea, natural phenomenon, or law of nature, and not for the abstract idea, natural phenomenon, or law of nature itself.

When considered "as a whole", examiner finds that neither the computer readable medium nor the system of claims 1 and 36 are "for a ***particular*** application of an abstract idea" [emphasis examiner's]. As appellants disclose in the specification, aspects of the claimed invention include: "identifying inferencing aspects for a program (pg. 2); including facts, rules or conclusions, (pg. 2); "rules" which are not those usually associated with AI but are used to make every day business decisions (pg. 7); frequent flyer miles and classifications thereof (pg. 8); people in China purchasing a camera (pg. 9-10). Therefore the examiner questions, what the particular practical application is. Can the appellant's invention really solve all problems known to man?

Applicants argue:

> Consider the method step of "evaluating, by the externalized inferencing component, the data comprising a set of rules to be interpreted against the parameters to perform an inference external to the program of instructions, wherein the externalized inferencing component is in communication with the program of instructions, wherein the inference is a

> derivation of the knowledge" (emphasis added), which comprises substantial limitations outside the realm of mere abstraction such as a mathematical formula without a practical application. While such a limitation may cover a board range of computer readable mediums and systems for executing a program of instructions, the claims clearly rise above the level of an abstract idea.

Examiner responds:

Appellants offer as “substantial limitations outside the realm of mere abstraction” a ‘method step of "evaluating, by the externalized inferencing component, the data comprising a set of rules to be interpreted against the parameters to perform an inference external to the program of instructions, wherein the externalized inferencing component is in communication with the program of instructions, wherein the inference is a derivation of the knowledge"’.

Examiner disagrees. What is claimed in the method step, “evaluating...by...externalized inferencing component...data comprising a set of rules to be interpreted against...parameters” is a 101 judicial exception of algorithm. While clearly embodied in hardware, the algorithm plus hardware (considered as either machine or computer related manufacture) must have a practical result. This is, however, not the case. Data comprising a

non-specific set of rules interpreted against a non-specific set of parameters is clearly abstract. Examiner asks: if the claim provides "substantial limitations outside the realm of mere abstraction", then in what specific and credible problem domain or realm (e.g., robot planning) does the claimed "derivation of...knowledge" occur? Claiming no specific problem domain as a limitation, the claimed "derivation of...knowledge" could be directed toward any conceivable realm, thus preempting the use of the claimed data abstractions for the derivation of knowledge in every conceivable area of human endeavor.

B. Rejection of Claims 1, 3-10, 13, 16-18, 20 and 36 Under 35 U.S.C. §102

Appellants' argue:

i. Claims 1, 3-10, 13, 16-18, 20 and 36

Claims 1 and 36 are the independent claims.

For a claim to be anticipated under 35 U.S.C. § 102, all elements of the claim must be found in a single prior art reference (see, e.g., Scripps Clinic & Research Found. v. Genentech Inc., 927 F.2d 1565, 1576, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991)). The identical invention must be shown in as complete detail as is contained in the claim. (See MPEP § 2131). The single prior art reference must disclose all of the elements of the claimed invention

functioning essentially in the same manner (see, e.g., Shanklin Corp. v. Springfield Photo Mount Corp., 521 F.2d 609 (1st Cir. 1975)).

In the case before the Board, Appellants respectfully assert that at the very least, the Final Action fails to present a legally sufficient basis for establishing a prima facie case of anticipation as against Claims 1, 3-10, 13, 16-18, 20 and 36: IBM teaches externalized business rules (see IBM, p. 32). IBM does not teach "passing the parameters to an externalized inferencing component upon executing a trigger point in the program of instructions" and "evaluating, by the externalized inferencing component, the data comprising a set of rules to be interpreted against the parameters to perform an inference external to the program of instructions" as claimed in Claims 1 and 36.

Examiner's response:

Examiner disagrees.

(A) Examiner asserts that IBM *does* teach "passing the parameters to an externalized inferencing component upon executing a trigger point in the program of instructions" as claimed in Claims 1 and 36. Examiner interpreted a classifier rule to be an externalized inferencing component (i.e., a Business Rule Bean (*see* pp. 2-4)). It should be noted that a classifier rule is only one of five types of rules IBM teaches as being supported in the BRBeans framework (*see* IBM, p. 3, §Different types of business rules). IBM clearly teaches that the rules are stored as persistent objects on a rule server (*see* IBM, p. 18, "• BRBeans EJBs: The WebSphere application server in which the BRBeans EJBs are installed is referred to as the BRBeans

rule server. These EJBs provide the underlying implementation for the business rule persistence...”). Examiner then interpreted as "passing the parameters to an externalized inferencing component upon executing a trigger point in the program of instructions" a situational trigger point passing a classifierPlist during execution of the tp.triggerSituational method (*see* IBM, pp. 26-27).

(B) Examiner asserts that IBM *does* teach "evaluating, by the externalized inferencing component, the data comprising a set of rules to be interpreted against the parameters to perform an inference external to the program of instructions" as claimed in Claims 1 and 36. Examiner interpreted “evaluating, by the externalized inferencing component, the data comprising a set of rules” to be *queries applied to the associated tables defined in WebSphere*. IBM clearly teaches that triggering the rules selected for firing requires accessing the rules in the database using parameters that are evaluated by a trigger point object (*see* IBM, p. 32, §Runtime behavior, “The first step in triggering a rule is for the trigger point framework to invoke a query method on the rule server to find the rules to be triggered. The main item used for the

query is the fully-qualified rule name. Other items used in the query include start and end date, whether or not this is a classifier, the classification of the rule, and whether or not the Rule is marked "ready".”). This can be clearly seen in the code fragment for the execution of the tp.triggerSituational method (*see* IBM, pp. 26-27) and in the sections on triggering (*see* IBM, p. 22, §Trigger Point Framework - overview, “2. Build the array of objects containing the runtime parameters needed to satisfy the trigger point's business purpose. This array is normally passed as one of the parameters of the fire method of the RuleImplementor. 3. Invoke the trigger ( ) , triggerClassifier ( ) , or triggerSituational ( ) method of the TriggerPoint class.”) and externalized business rules (*see* p. 32, §Externalized business rules).

Appellants’ argue:

The externalized business rules taught by IBM make a classification based on logic and data.

Examiner’s response:

A classifier rule is only one of five types of rules IBM teaches as being supported in the BRBeans framework (*see* IBM, p. 3, §Different types of business rules).

Appelants' argue:

> Respectfully, making a classification based on logic and data is not analogous to the externalized inferencing component claimed in Claims 1 and 36. Nowhere does IBM teach inferencing which derives new knowledge from rules and knowledge (data and parameters). Rather, IBM teaches business rules that simply return a "value." Returning a value is not analogous to making an inference. For example, consider IBM's maxTruckGrossWeight rule (see IBM, p. 33). This business rule, which checks whether a truck weight is valid, compares a truck weight entered by a user with a single, predetermined value representing a maximum truck weight. The rule determines whether the value entered by the user is above or below the maximum truck weight and returns an appropriate value.

Examiner's response:

Appellants provide IBM's "maxTruckGrossWeight rule" (*see* IBM, p. 33) as an example of why IBM doesn't teach inferencing as deriving new knowledge from rules and knowledge. Appellants assert that "This business rule, which checks whether a truck weight is valid, compares a truck weight entered by a user with a single, predetermined value representing a maximum truck weight. The rule determines whether the value entered by the user is above or below the maximum truck weight and returns an appropriate value." Appellants' conclusion ignores IBM's description of their inference procedure.

IBM states, on pg. 33: "MaxRuieImpl, when invoked, tests the parameter it is passed against the initialization list value and returns a ConstraintReturn. The ConsraintReturn is set to true if the passed parameter is less than or equal to the initialization value. Otherwise, the ConstraintReturn is set to false "and some information is added describing which values were compared and why the test failed." Clearly, a "ConstraintReturn" object, comprising (i) a Boolean state of true or false plus (ii) "information added describing which values were compared and why the test failed", is not just a *value* (e.g., a string, Boolean value, integer, real number, etc.) as it comprises multiple *types* of values (i.e., a Boolean and a string).

Appelants' argue:

Nowhere does the rule make an inference deriving new knowledge. Thus, IBM does not teach inferencing which derives new knowledge, essentially as claimed in Claims 1 and 36.

Examiner's response:

At this juncture, appellants raise the central questions of their 102 arguments:

What is ***new*** knowledge?

Does the prior art teach ***inferencing new*** knowledge?

What is deriving ***new*** knowledge, ***essentially as claimed*** in Claims 1 and 36?

We may also ask if knowledge is ***new*** if it is entailed by existing rules?

First, appellants provide no ***definition*** of new knowledge. They do provide a paradoxical framing of what they may mean by new knowledge at para. [0032]:

> [0032] In the context of externalization, "rules" are not those usually associated with the artificial intelligence community, but are rather ones used to make everyday "business" decisions. The technique employed is more structurally oriented than declarative, and the rules employed are often straightforward. In general, new knowledge is not sought after, but instead time and situational variability is easily managed.

But, what does this mean? If “new knowledge is not sought after”, why the emphasis on new knowledge and how do appellants' justify the assertion that “IBM does not teach inferencing which derives new knowledge, essentially as claimed in Claims 1 and 36.” What does it mean that “instead time and situational variability is easily managed”? What does that have to do with new knowledge?

While appellants do not define new knowledge, appellants do provide an example:

> [0039] Typically, reasoning or knowledge based systems can be used to learn new facts. For example, it might be learned that when people in China purchase a camera, they often also purchase a carrying case; whereas people in France may purchase batteries in addition to a camera.

However, examiner asserts that such findings are usually the result of data mining which generates rules representing the findings using statistical techniques. Such new knowledge is *not* typically a result of deduction, either externalized or from rules programming.

Examiner, thus, finds appellants silent as to what new knowledge is. Further, appellants are silent on the well known techniques of augmenting rule based systems with the capability to revise conclusions (e.g., circumscription, default reasoning, autoepistemic logic, abductive reasoning, etc.) (*see* TOMPITS, "A Survey of Non-Monotonic Reasoning", 1995 and Inoue, "Automated Abduction", 2002).

We may note that IBM does disclose a type of application of abductive reasoning in providing an explanation of the fact (known after the return of the ConstraintReturn) that the test failed (see pg. 33).

Examiner asserts that the "information added describing which values were compared and why the test failed" returned by the "ConstraintReturn" object ***is*** new knowledge unless the system has stored this information before executing the "maxTruckGrossWeight rule". More importantly: unless the system has stored this information for *any* particular test (thus *all* particular tests) *before* executing the "maxTruckGrossWeight rule" for a particular weight, the "information added describing which values were compared and why the test failed" is necessarily "new knowledge", as far as the system is concerned. Appellants are silent as to whether the new knowledge is new to system users or the system itself. However, Examiner finds that it is quite clear that IBM's externalized rules derive new knowledge.

Appellants' invention provides an approach to the manipulation of rules. Examiner finds the core of the invention disclosed in paragraphs [0071]-[0085] of the specification. Examiner understands appellants' invention to compose new knowledge by performing various re-writes to rule structures by "dynamic mappings" (*see* specification,

[0083]) and static and dynamic compositions (*see* specification, [0084]) under the control of applications with trigger points, within, that employ externalizable inference components (EICs) to control the processes described in paragraphs [0071]-[0085].

> [0051] The present invention allows for placement of trigger points within applications that employ externalizable inference components (EICs)...Typically, applications will pass context and parameter information to trigger points, which then dynamically identify and employ EICs.

If we contrast appellants' disclosure to, what IBM discloses:

"BRBeans Rule is a persistent object that exists on the BRBeans Rule server" (*see* IBM, p. 29); and

"(reusing) and exercising various components in new ways" (*see* IBM, §Writing your own rule implementors, p. 31); and

with the fire method: "Any desired algorithm can be performed here"; and

the BRBeans, rule administration allows "creating new rules that didn't exist before" (*see* IBM, p. 16, Part III: Rule Administration, §Rule administration); and

rule administration provides a "programmatic interface that can be used by programmers writing code to manage rules" (*see* IBM, p. 16, Part III: Rule Administration, §Rule administration)

we see clearly that IBM, at least, suggests reusing and exercising the externalized business rules in new ways using the Java code.

In contrast, appellants disclose prior art with externalized business rules as well as rule re-writing (*see* specification, [0004]) and (*see* Moore et al. USPN: 5,446,885, SUMMARY OF THE INVENTION, "(15) The rules are stored in a database as objects." and "(16) Another aspect of the system allows a user to modify existing rules and create new rules."). Then, appellants assert that "the prior art does not disclose the use of externizable inferencing components", presumably EICs, (*see* specification, [0004]) which they provide as "trigger points" (*see* specification, [0006]). Now, EICs are, at least, analogous to what IBM provides (*see* IBM, p. 22, Trigger Point Framework - overview, a "trigger point is simply the location in a method of an object at which externalized business rules are invoked" and "a trigger point is placed in user-written code").

So, how is it that "IBM does not teach inferencing which derives new knowledge, essentially as claimed in Claims 1 and 36"? Perhaps appellants mean that IBM doesn't

disclose or suggests the procedures described by appellants in paragraphs [0071]-[0085] of the specification. However, these procedures are not what appellants claim. What appellants claim is EICs and trigger points.

In summary, since appellants haven't defined *new* knowledge (and thus inferencing new knowledge) and appellants did not ***claim*** the novel aspects of their approach to deriving *new* knowledge we may safely concluded that IBM *does* teach inferencing new knowledge, essentially *as claimed* in Claims 1 and 36.

Appellants' argue:

In formulating the rejection of Claims 1 and 36 under 35 U.S.C. § 102, the Examiner relies on an interpretation that the Classifier Trigger Point of IBM teaches that a value is new knowledge (see Final Action, p. 25). Applicants respectfully disagree. The Classifier Trigger Point taught by IBM merely makes a classification based on logic and data. Consider the example of a Classifier Trigger Point given on page 25 of IBM. The Classifier Trigger Point receives data (e.g., customers' spending history) and uses this data to classify the customers into different levels (e.g., Gold, Silver and Bronze). The Classifier Trigger Point clearly does not make an inference, nor does it derive new knowledge. Rather, it merely classifies customers into different levels based on logic and data (e.g., if customer spent less than X, classify customer as Bronze; if customer spent more than X but less than Y, classify customer as Silver; if customer spent more than Y, classify customer as Gold).

Examiner's response:

Examiner disagrees. IBM's Classifier Trigger Point performs the same function in application code as does a trigger point in appellants' invention (*see* arguments above). Further, a Classifier Trigger Point can execute any desired algorithm and create "new rules that didn't exist before" (*see* arguments above) just like the base rule types. Clearly, "new rules that didn't exist before" created by Java code called from a Classifier Trigger Point can derive "new knowledge" (assuming that "new knowledge" is not entailed by the existing rules).

Appellants' argue:

Externalized business rules, such as the Classifier Trigger Point taught by IBM, and reasoning systems that employ inferencing, are discussed on pages 7-10 of the present Application. The Application clearly distinguishes between these two models, stating that "new knowledge is not sought" in the context of externalization (see Spec., page 7, line 20 - page 8, line 2). To illustrate that externalized business rules do not seek new knowledge, an example analogous to IBM's Classifier Trigger Point example (IBM, p. 25) is given (see Spec., page 8, lines 3-12 - e.g., a rule classifying a frequent flier to be bronze, silver, or gold based upon the number of miles flown with the airline during one year is an externalized business rule and does not seek new knowledge). Accordingly, externalized business rules such as IBM's Classifier Trigger Point merely make a classification based on logic and data, and do not perform inferencing which derives new knowledge from rules and knowledge (data and parameters), essentially as claimed in Claims 1 and 36. Thus, IBM does not teach inferencing which derives new knowledge, essentially as claimed in Claims 1 and 36.

Examiner's response:

Examiner does not understand how '"new knowledge is not sought" in the context of externalization' distinguishes externalized business rules, such as the Classifier Trigger Point taught by IBM, and applications (e.g., reasoning systems) that employ inferencing from appellants' externalized rules. Examiner does not see how appellants conclude from the disclosed example of classification that "externalized business rules such as IBM's Classifier Trigger Point merely make a classification based on logic and data, and do not perform inferencing which derives new knowledge from rules and knowledge (data and parameters), essentially as claimed in Claims 1 and 36." Examiner requests a detailed explanation of how this conclusion is arrived at.

Appellants' argue:

Further, the Examiner interprets the dependent rules taught by IBM (see IBM, p. 21) as teaching "that inferencing derives new knowledge (especially used to classify situations) from rules and knowledge (data and parameters)." (See Final Action, p. 25). Applicants respectfully disagree. Although the Examiner asserts that IBM's dependent rules teach that inferencing derives new knowledge, it is never explained how this is done. IBM teaches dependent rules, which are rules triggered by other rules. Regardless of the manner in which these rules are triggered, they are still externalized business rules that do not seek new knowledge, and instead merely make a classification based on logic and data. Every dependent rule called upon by another rule is simply returning a value; no inference is ever made. Thus, the dependent rules of IBM do not teach inferencing that derives new knowledge.

Examiner's response:

Examiner notes that IBM's dependent rules can generate new knowledge when the result from a series of dependent rule firings is a result that was unknown to the system before the firing of the series of dependent rules. IBM teaches "Separating rules from the business logic of the application makes it easy to reuse a business practice decision in a consistent fashion." (*see* IBM, §Why externalize rules?, p. 4). IBM also teaches that with the fire method: "Any desired algorithm can be performed here." (*see* IBM, §Writing your own rule implementors, p. 31). Further, IBM teaches that BRBeans rule administration allows "creating new rules that didn't exist before" (*see* IBM, p. 16, Part II1: Rule Administration, §Rule administration). In addition, IBM teaches that rule administration provides a "programmatic interface that can be used by programmers writing code to manage rules" (*see* IBM, p. 16, Part II1: Rule Administration, §Rule administration). Thus, examiner fails to see why dependent rules in the BRBeans framework cannot be used to derive new knowledge.

Appellants' argue:

The Examiner's rejection under 35 U.S.C. § 102 further relies on an interpretation that an IF/THEN statement, as taught by *IBM*, is analogous to the claimed inferencing which derives new knowledge (see Final Action, p. 26-27). Applicants respectfully disagree. IBM teaches an IF/THEN statement that is used only in classification.

Examiner's response:

Examiner disagrees. IBM teaches:

> At its simplest level, a business rule is little more than a well placed IF/THEN statement that compares a variable against a determined value, and then issues a command when they match. (*see* IBM, p. 2, §What is a business rule?)

Examiner asserts that the command issued when a variable matches a determined value is not constrained to classification (*see* IBM, pp. 2-3, Examples 1-2). Only in Example 3 does the command perform a classification. Whether an IF/THEN statement derives "new knowledge" is completely dependent upon whether the knowledge (consequent of the IF/THEN) is stored by the system before execution of the IF/THEN statement.

Appellants' argue:

> IBM does not teach "evaluating, by the externalized inferencing component, the data comprising a set of rules to be interpreted against the parameters to perform an inference external to the program of instructions, wherein the inference is a derivation of the knowledge" (emphasis added) as claimed in Claims 1 and 36. The IF/THEN statement of IBM is a conditional statement that lacks inherent inferencing capabilities. That is, the IF/THEN statement of IBM is merely a logical argument that cannot be considered an inference as it, by itself, does not derive knowledge. Indeed, it is instructive that IBM does not use the word "inference" or its derivatives.

Examiner's response:

If an IF/THEN statement is considered by appellants to lack "inherent inferencing capabilities", then examiner is uncertain as to what appellants mean by "inherent inferencing capabilities". The IF/THEN statement is well

known to implement *the* basic inference rule, called modus ponens (*see He et al.*, "CS381 Discrete Structures/Discrete Mathematics Web Course Material", §Reasoning with Propositions, 'For example if "if it rains, then the game is not played" and "it rains" are both true, then we can conclude that the game is not played.'). Inherent inferencing capabilities are neither claimed nor defined in the specification by appellants therefore, examiner will assume that "inherent inferencing capabilities" are those properties defined for the term as it is commonly used, i.e., "The basic inference rule is modus ponens. It states that if both P->Q and P hold, then Q can be concluded…" (*see* CS381 Discrete Structures/Discrete Mathematics Web Course Material, Reasoning with Propositions).

Examiner also calls attention to the fact that appellants' method, disclosed in paragraphs [0071]-[0085], implicitly uses IF/THEN statements

> [0084] A composition, such as EIC 840, can occur statically (prior to runtime) or dynamically (at runtime). "Rule:1(p(x0))" represents "if condition is `condition function p(x0)` then result is `result x0`". Similarly, "Rule:2(q(x0))" represents "if condition is `condition function q(x0)` then result is `result x0`".
>
> [0085] More concretely, "Rule:3(c)" might represent "if customer status `c` then give customer discount `c`"; "c->r(y0)" might represent "substitute `condition: bronze, result: lookupPercentage(bronze)` for `c`"; the combination

> results in: if customer has status `bronze` then give customer discount `looked-up percentage for bronze`.

If IF/THEN statements are considered by appellants to lack "inherent inferencing capabilities", then why do appellants' forms represent them?

Appellants' argue:

> Further, consider that the claimed inferencing component of Claims 1 and 36 is comprised of more than a mere IF/THEN statement (see FIG. 5 of the present application, illustrating an inferencing component comprised of multiple rule sets and algorithms).

Examiner's response:

Appellants do not *claim* the Rule Set Component (RSC) of FIG. 5. Appellants only claim an "inferencing component" (i.e., trigger points) which is anticipated by IBM's BRBeans trigger point.

Appellants' argue:

> Thus, the IF/THEN statement of IBM, on its own, lacks the complexity for deriving knowledge. In view of the foregoing, the teachings of IBM are not analogous to an externalized inferencing component performing an inference external to the program of instructions, wherein the inference is a derivation of knowledge, essentially as claimed in Claims 1 and 36.

Examiner's response:

Examiner disagrees. First, the IF/THEN statement of IBM, does not act on its own. It acts within the BRBeans

Trigger Point Framework. Second, IBM's BRBeans rules are more complex than just the IF/THEN statements IBM uses as a simplified example on pp. 2-3 (*see* IBM, p. 3, §Different types of business rules, e.g., "**Invariant rules:** A rule that ensures that multiple changes made by an operation are properly related to one another."). Third, examiner finds that the BRBeans framework can combine both externalized rules, EJB externalized components, and the capability for the rule developer to implement firing methods that can perform any algorithm implementable in Java (i.e., in any third generation programming language). Thus, BRBeans is analogous to an externalized inferencing component performing an inference external to the program of instructions, wherein the inference *is* a derivation of knowledge, essentially as claimed in Claims 1 and 36.

Appellants' argue:

Therefore, IBM, fails to teach all of the limitations of Claims 1 and 36.

Claims 3-10, 13, 16-18 and 20 depend from Claim 1. The dependent claims are believed to be allowable for at least the reasons given for Claim 1.

Examiner's response:

Examiner considers IBM to teach all of the limitations of claims 1 and 36 and claims 3-10, 13, 16-18 and 20 which depend from claim 1. The dependent claims are *not* believed to be in a condition for allowance for at least the reasons given for Claim 1.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Nathan H. Brown, Jr./

Examiner, Art Unit 2129

Conferees:

/David R Vincent/

Supervisory Patent Examiner, Art Unit 2129

/Michael B. Holmes/

Primary Examiner, Art Unit 2129